Claim Amendments:

Please amend the claims as indicated:

1. (original) A method comprising the steps of:
receiving a first request to access data from a first memory device;
preparing the first request for the data for access through the first memory device;
providing a second request to access the data from a second memory device, wherein the second request is provided concurrently with the step of preparing the first request;
receiving a first notification that the data associated with the first request is available from the second memory device; and
terminating the first request, in response to the first notification.

A1

2. (original) The method as in Claim 1, wherein terminating the first request includes terminating the first request in a memory controller before the first request is sent to the first memory device.

3. (original) The method as in Claim 1, wherein terminating the first request includes terminating data received from the first memory device, wherein the data is associated with the first request.

4. (original) The method as in Claim 3, wherein the data is terminated in a memory controller.

5. (original) The method as in Claim 1, wherein the data in the second memory device is coherent with the data in the first memory device.

6. (original) The method as in Claim 1, wherein the first memory device includes random access memory.

7. (original) The method as in Claim 1, wherein the second memory device includes cache memory.

8. (original) The method as in Claim 1, wherein a memory controller associated with the first memory device terminates the memory request in response to the termination request.

9. (original) The method as in Claim 1, wherein the first request is generated by a client on a system bus.

10. (original) The method as in Claim 1, wherein the memory request includes a multiple target memory request.

11. (original) The method as in Claim 1, wherein the step of providing a second request includes:
providing the second request to a bus interface unit; and
wherein the bus interface unit is coupled to the second memory device.

12. (original) A method comprising the steps of:
receiving a first request to read data from a memory device;
preparing a second request, based upon the first request, for transmission to the memory device;
delivering a third request, based upon the first request, for data from a cache memory, the third request being delivered concurrently with the preparation of the second request;
providing, in response to the first request, data from the cache memory when the data stored in the cache memory is coherent with the data stored in the memory device;
terminating the second request when the data is provided from the cache memory; and
providing, in response to the first request, data from the memory device when the data stored in the cache memory is not coherent with the data stored in the memory device.

13. (original) The method as in Claim 12, wherein the step of terminating the second request further includes stopping delivery of the second request to the memory device.

14. (original) The method as in Claim 12, wherein the step of terminating the second request includes further stopping data from the memory device from being provided in response to the first request.

15. (original) The method as in Claim 12, wherein a memory controller associated with the memory device is used for the step of terminating the second request.

16. (original) The method as in Claim 12, wherein the first request is generated by a bus client.

A1

17. (original) The method as in Claim 15, wherein the client is a multiple target memory request.

18. (original) The method as in Claim 12, wherein delivering a third request includes:
delivering the third request to a bus interface; and
wherein the bus interface unit is coupled to the cache memory.

19. (original) A system comprising:
a data processor having:
an input/output buffer; and
cache memory to store data associated with a memory device;
a bus interface unit having a first input/output buffer coupled to the input/output buffer of the data processor, a second input/output buffer and a third input/output buffer, said bus interface unit to:
determine a validity of data in said cache memory during a cache access; and
provide a notification indicating data in said cache memory is valid, wherein said notification identifies a first request;
said memory device having an input/output buffer coupled, said memory device to provide data associated with a first request;
a bus controller having a first input/output buffer coupled to the input/output buffer of the data processor, a second input/output buffer coupled to the second input/output buffer of the bus interface unit and a third input/output buffer, said bus controller to:

receive said first request to access data in said memory device, wherein said request is received from a bus client;

provide said first request to the memory controller;

receive said data associated with said first request from said bus interface unit;

the memory controller having a first input/output buffer coupled to the third input/output buffer of the bus controller, a second input/output buffer coupled to the third input/output buffer of the bus interface unit and a third input/output buffer coupled to the input/output buffer of the memory device, said memory controller to:

provide access to said memory device;

receive said first request from said bus controller;

prepare said first request to access data from said memory device;

provide a second request to said bus interface unit, wherein said second request is to access data associated with said first request from said cache memory;

receive said notification from said bus interface unit; and

terminate the first request, in response to the receipt of said notification.

20. (original) The system as in Claim 19, wherein said bus interface unit further used to synchronize said cache memory to said memory device.

21. (original) The system as in Claim 20, wherein said bus interface unit to determine validity includes determining a coherency between said cache memory and said memory device.

22. (original) The system as in Claim 21, wherein coherency is dependent on whether said memory device has been written to prior to a synchronization of said cache memory with said memory device.

23. (original) The system as in Claim 19, wherein said bus controller includes a peripheral component interconnect bus controller.

24. (original) The system as in Claim 19, wherein said memory device includes random access memory.

25. (original) The system as in Claim 19, wherein said memory controller further used to:
assign a first identifier to said first request; and
identify said first request from a plurality of pending requests using said first identifier.

26. (original) The system as in Claim 19, said memory controller further used to:
generate a second identifier using said notification; and
storing said identifier as part of a kill list, wherein said kill list identifies requests to be terminated.

A1

27. (original) The system as in Claim 19, wherein said memory controller to terminate the first request includes terminating data received from said memory device, wherein said data is associated with said first request.

28. (new) A system comprising:
a cache memory;
a memory device; and
a memory controller configured to:
provide access to said memory device;
receive a first request to access data in said memory device;
prepare said first request to access data from said memory device;
provide a second request to access data associated with said first request from said cache memory;
receive a notification indicating data in said cache memory is valid, wherein said notification identifies the first request; and
terminate the first request, in response to the receipt of said notification.